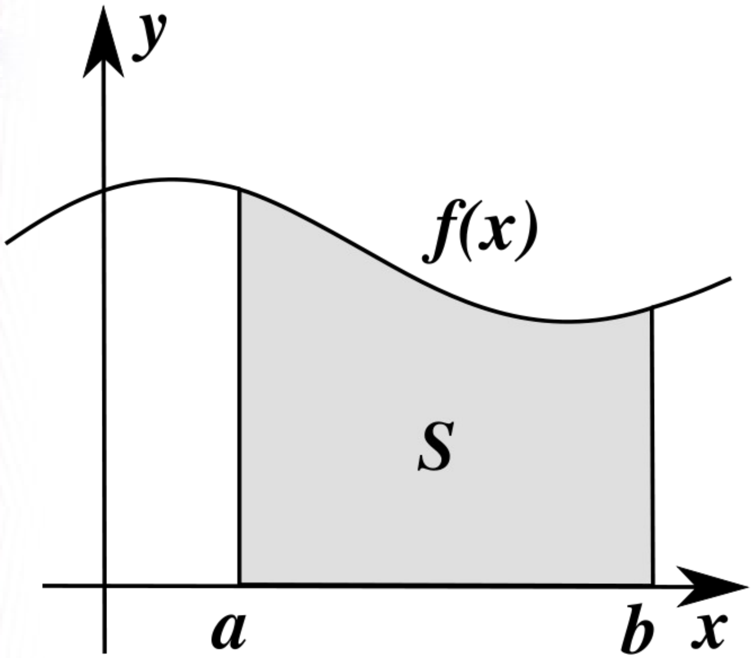
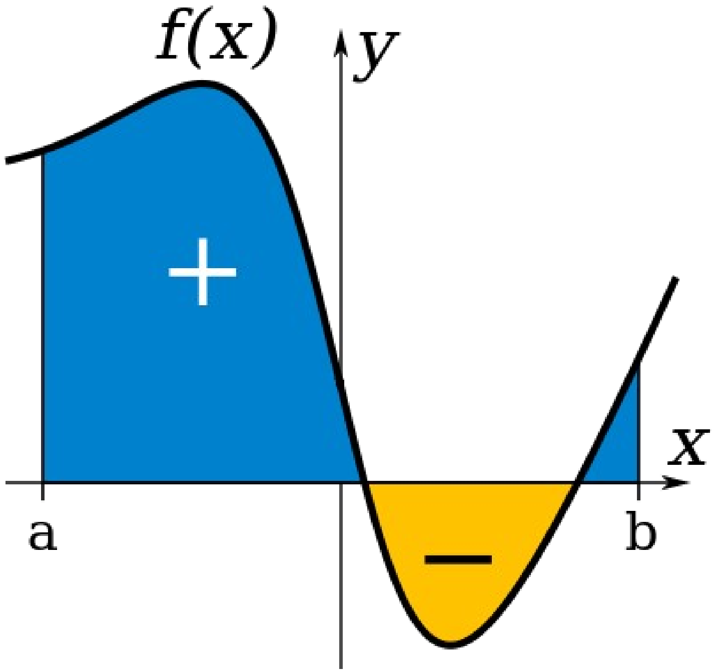
Definite Integral using Monte-Carlo Method

Febrie Ahmad Azizi - 20912008
Ridlo Wahyudi W. - 20912009

2012 – Computational Science

Integral Fungsi Sembarang - 1D

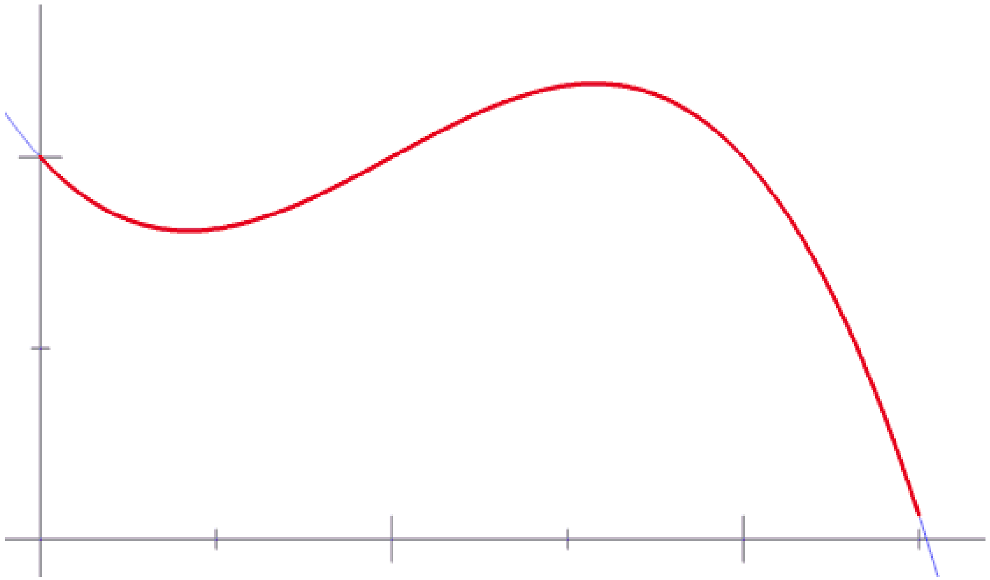




Analitik :

$$\int_a^b f(x)\,dx = F(b) - F(a)$$ → Luas S

Numerik :



Riemann Integral

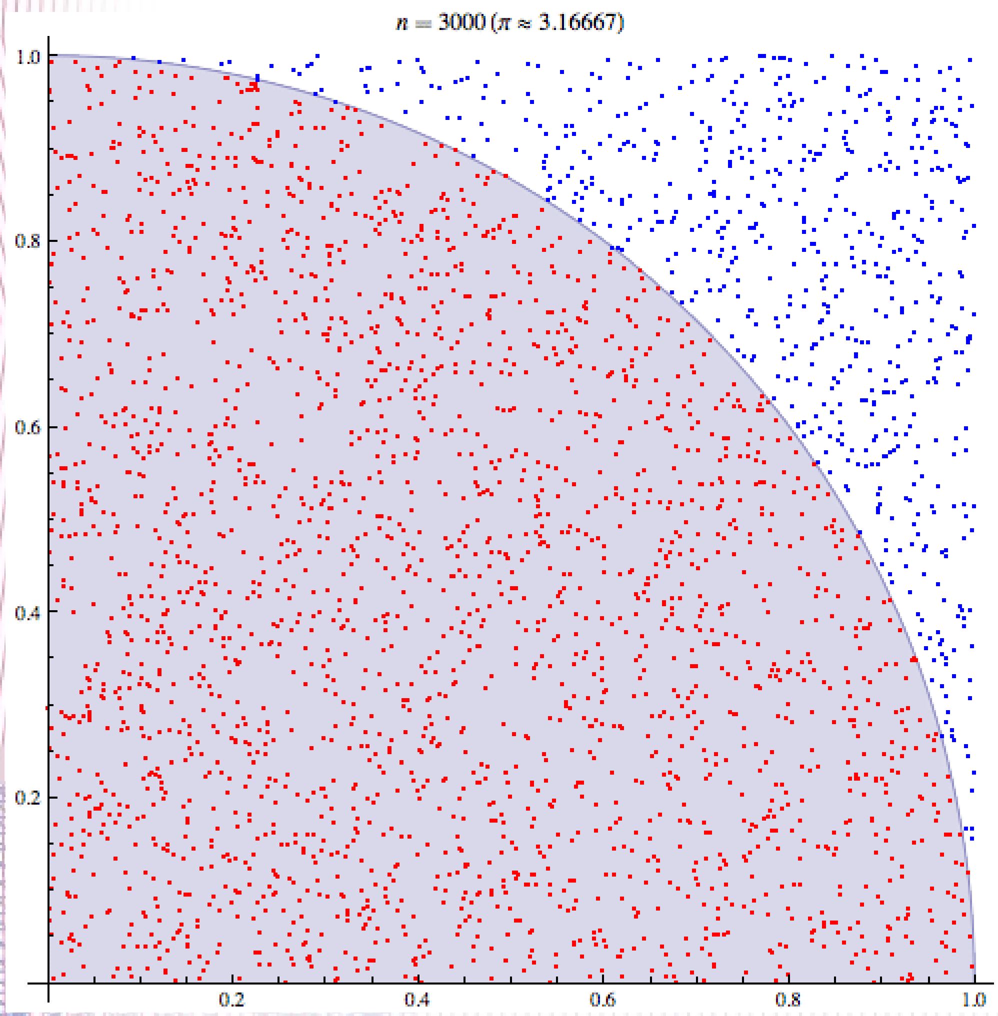
Banyak metode, antara lain:
- Newton-Cotes Formulae → Rectange rule, Trapezoid rule, Simpson's 1/3, Simpson's 3/8
- Romberg Integration (Extrapolation), dengan h berbeda.
- Gaussian Quadrature, dll

Penerapan metode ini untuk integrasi dimensi tinggi (multiple integral) terbatas

Monte Carlo Method

Monte Carlo methods (or Monte Carlo experiments) are a class of computational algorithms that rely on repeated random sampling to compute their results.

Approximate PI



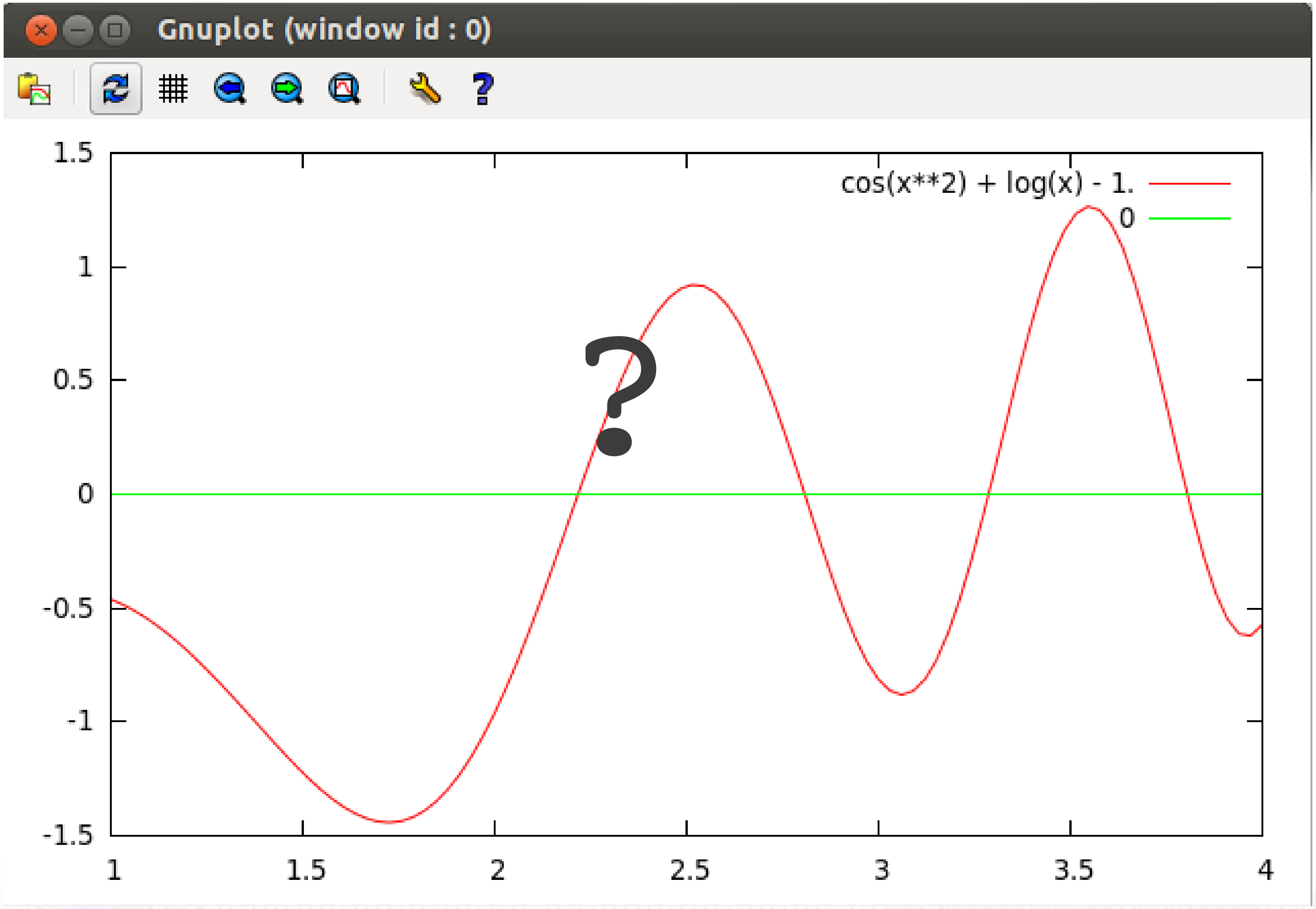
Algorithm:

```
FOR i=1, Ntot
generateRandom (0 <= r1 <= 1 ; 0
<= r2 <= 1)
IF (r1*r1 + r2*r2 <= 1)
    N += 1
ENDFOR

Result_PI = N/Ntot * 4
```

Monte-Carlo untuk integral fungsi sembarang 1D

→ Sama dengan kasus mencari PI



Algorithm:

1. Cari maksimum/minimum dari fungsi f(x) pada selang **A** dan **B** → **C**
(A dan B → batas integrasi)

2. Lakukan hingga **Ntotal**:
- **random** titik untuk **X** dengan batas **A** sampai **B**
- periksa apakah nilai **f(X**) positif atau negatif
    Jika **f(X) positif** :
        + **random** untuk **Y** dengan batas **0** sampai **C**
        + periksa apakah **Y** berada **di bawah** kurva **f(x)**, jika iya **N = N+1**
    Jika **f(X) negatif** :
        + **random** untuk **Y** dengan batas **0** sampai **-C**
        + periksa apakah **Y** berada **di atas** kurva **f(x)**, jika iya **N = N-1**

3. **Hasil = N/Ntot*c*(b-a)**

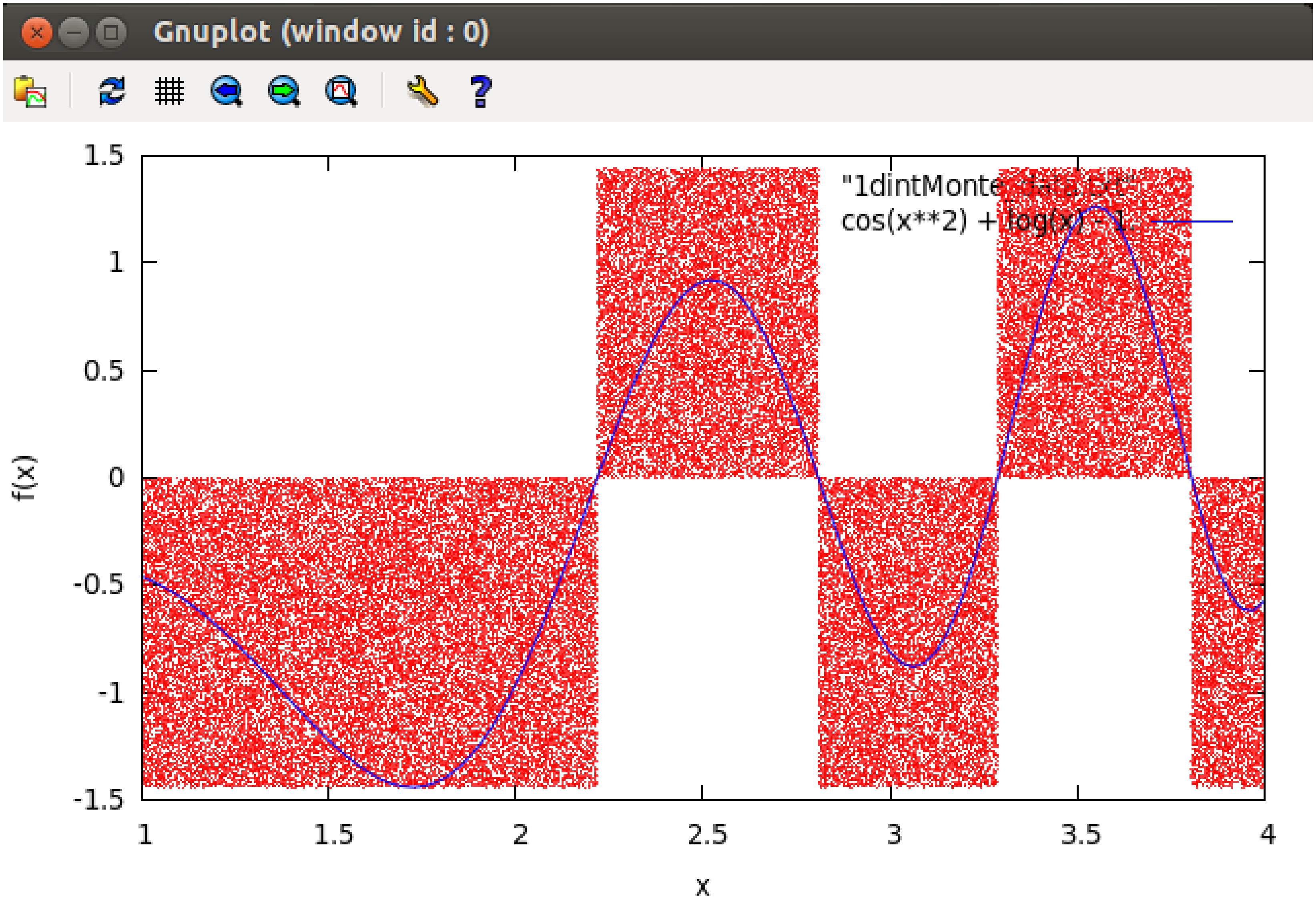
Code :

```
//mencari maximum fungsi dengan cara monte-carlo
 for (i=0;i<Ntot;i++){
     x = a + (b-a)*unirand();
     if (fabs(f(x)) > c) {c = fabs(f(x));}
 }

 //mencari integral fungsi dengan cara monte-carlo
 for (i=0;i<Ntot;i++){
     x = a + (b-a)*unirand();
     if (f(x) > 0.){
          y = c*unirand();
          if (y <= f(x)){ N++; }
     }
     else {
          y = -c*unirand();
          if (y >= f(x)){ N--; }
     }
 }

 res = (double)N / (double)Ntot * c*(b-a);
```

Result :



Untuk kurva berkelakuan baik pada batas integrasi, hasilnya cukup baik

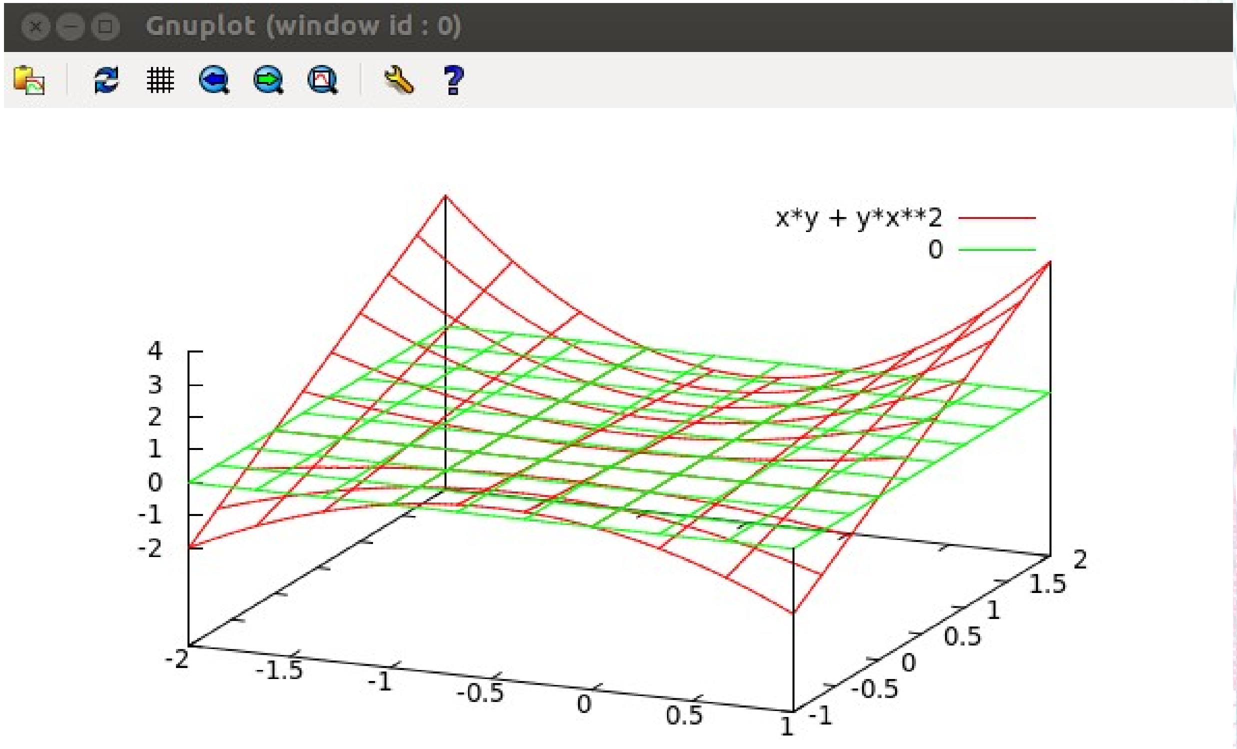
Integral untuk dimensi yang lebih tinggi dengan Monte-Carlo

→ di tambah parameter random sesuai jumlah parameter di dalam fungsi

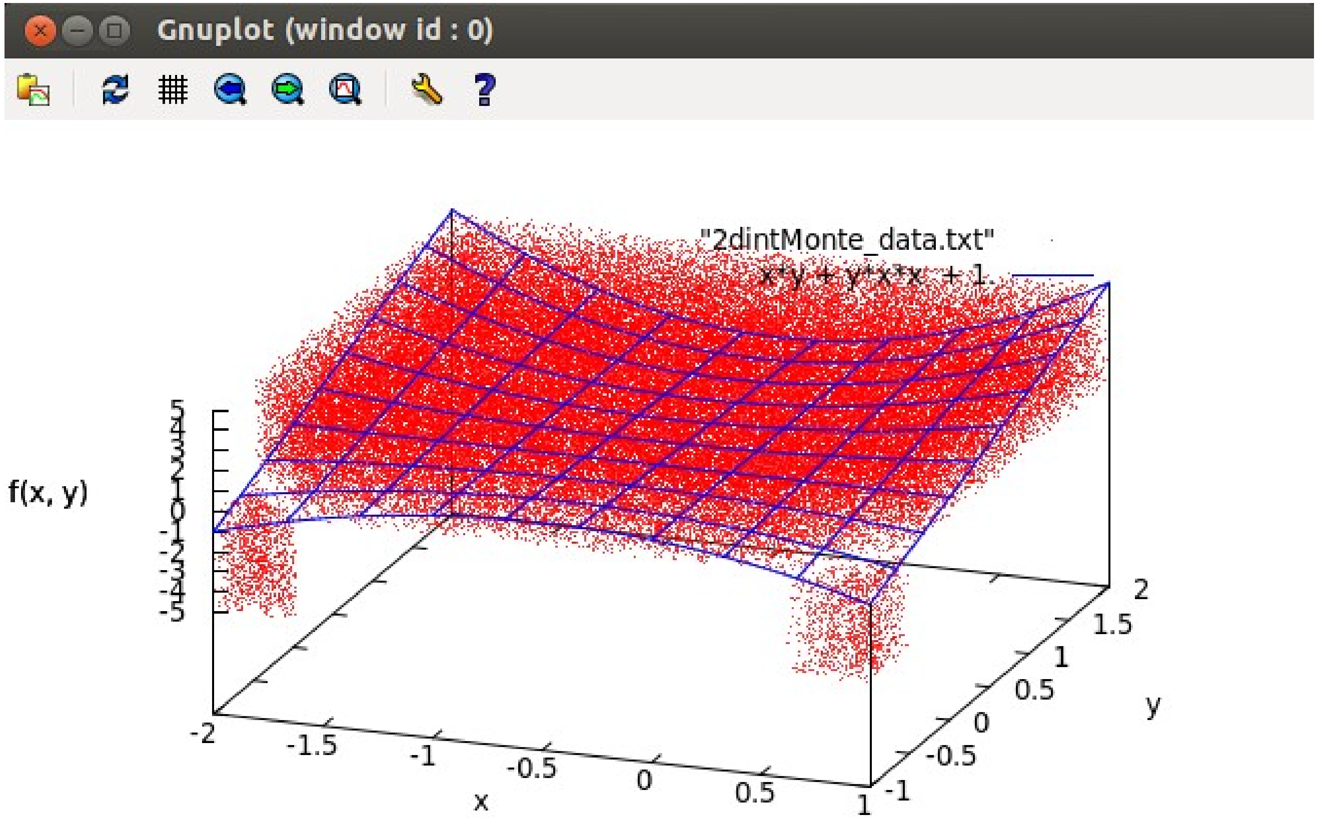
Contoh untuk 2D / orde 2:

$$\int_{-1}^{2}\int_{-2}^{1}\left(xy+yx^{2}\right)dxdy$$

Exact Solution : 11.25



Monte-Carlo Result :



N total	10000	100000	1000000
run_1	11.498472	11.349053	11.257513
run_2	11.309235	11.169831	11.220483
run_3	11.220745	11.253155	11.252168
average	11.342817	11.257346	11.243388
stdev	0.1418763	0.0896844	0.0200155

Orde 3

$$\int_1^4 \int_0^2 \int_1^3 \left(xyz^2 + x^2 yz + xy^2 z^2 \right) dxdydz$$

Exact Solution : 522

Monte-Carlo Result:

N total	10000	100000	1000000
run_1	517.770038	518.891995	521.515282
run_2	529.565725	523.920987	522.052511
run_3	527.928108	520.128073	521.862817
average	525.087957	520.980351	521.810203
stdev	6.39018006	2.62058663	0.27245164

Simpulan

1. Metode Monte Carlo mudah diterapkan untuk kasus integral lipat banyak (multiple integral).

2. Semakin banyak jumlah titik random, maka semakin akurat hasil yang diperoleh.

2. Untuk permasalahan kompleks dapat dilakukan pembagian part integrasi dan melakukan perhitungan dengan menggunakan banyak komputer sekaligus (parallel computing).

3. Pelopor metode heuristic untuk memecahkan kasus-kasus lain.

Thank You, ^^